ATTACHMENT B

Subnetting

Previous  Next

Subnetting an IP Network can be done for a variety of reasons, including organization, use of different physical media (such as Ethernet, FDDI, WAN, etc.), preservation of address space, and security. The most common reason is to control network traffic. In an Ethernet network, all nodes on a segment see all the packets transmitted by all the other nodes on that segment. Performance can be adversely affected under heavy traffic loads, due to collisions and the resulting retransmissions. A router is used to connect IP networks to minimize the amount of traffic each segment must receive.

Subnet Masking

Applying a subnet mask to an IP address allows you to identify the network and node parts of the address. The network bits are represented by the 1s in the mask, and the node bits are represented by the 0s. Performing a bitwise logical AND operation between the IP address and the subnet mask results in the *Network Address* or Number.
For example, using our test IP address and the default Class B subnet mask, we get:

```
10001100.10110011.11110000.11001000      140.179.240.200   Class B IP Address
11111111.11111111.00000000.00000000      255.255.000.000   Default Class B Subnet Mask
---------------------------------------------------------
10001100.10110011.00000000.00000000      140.179.000.000   Network Address
```

Default subnet masks:

- **Class A** - 255.0.0.0 - 11111111.00000000.00000000.00000000
- **Class B** - 255.255.0.0 - 11111111.11111111.00000000.00000000
- **Class C** - 255.255.255.0 - 11111111.11111111.11111111.00000000

Previous  Next

Updated January 29, 2007

ATTACHMENT C

TCP / IP Suite

IP

RFC 791 http://www.cis.ohio-state.edu/htbin/rfc/rfc791.html

The Internet Protocol (IP), defined by IETF RFC791, is the routing layer datagram service of the TCP/IP suite. All other protocols within the TCP/IP suite, except ARP and RARP, use IP to route frames from host to host. The IP frame header contains routing information and control information associated with datagram delivery.

The IP header structure is as follows:

▲TOP

4	8	16	32 bits
Ver.	IHL	Type of service	Total length
Identification		Flags	Fragment offset
Time to live	Protocol	Header checksum	
Source address			
Destination address			
Option + Padding			
Data			

IP header structure

Version

Version field indicates the format of the Internet header.

IHL

Internet header length is the length of the Internet header in 32-bit words. Points to the beginning of the data. The minimum value for a correct header is 5.

Type of service

Indicates the quality of service desired. Networks may offer service precedence, meaning that they accept traffic only above a certain precedence at times of high load. There is a three-way trade-off between low delay, high reliability and high throughput.

Bits 0-2: Precedence
111 Network control.
110 Internetwork control.
101 CRITIC/ECP.
100 Flash override.
011 Flash.
010 Immediate.
001 Priority.
000 Routine.

Bit 3: Delay

0 Normal delay.
1 Low delay.

Bit 4: Throughput

0 Normal throughput.
1 High throughput.

Bit 5: Reliability

0 Normal reliability.
1 High reliability.

Bits 6-7: Reserved for future use.

Total length
Length of the datagram measured in bytes, including the Internet header and data. This field allows the length of a datagram to be up to 65,535 bytes, although such long datagrams are impractical for most hosts and networks. All hosts must be prepared to accept datagrams of up to 576 bytes, regardless of whether they arrive whole or in fragments. It is recommended that hosts send datagrams larger than 576 bytes only if the destination is prepared to accept the larger datagrams.

Identification
Identifying value assigned by the sender to aid in assembling the fragments of a datagram.

Flags
3 bits. Control flags:

Bit 0 is reserved and must be zero

Bit 1: Don't fragment bit:

0 May fragment.
1 Don't fragment.

Bit 2: More fragments bit:

0 Last fragment.
1 More fragments.

Fragment offset
13 bits. Indicates where this fragment belongs in the datagram. The fragment offset is measured in units of 8 bytes (64 bits). The first fragment has offset zero.

Time to live
Indicates the maximum time the datagram is allowed to remain in the Internet system. If this field contains the value zero, the datagram must be destroyed. This field is modified in Internet header processing. The time is measured in units of seconds. However, since every module that processes a datagram must decrease the TTL by at least one (even if it processes the datagram in less than 1 second), the TTL must be thought of only as an upper limit on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded and to bound the maximum datagram lifetime.

Protocol
Indicates the next level protocol used in the data portion of the Internet datagram.

Header checksum
A checksum on the header only. Since some header fields change, e.g., Time To Live, this is recomputed and verified at each point that the Internet header is processed.

Source address / destination address
32 bits each. A distinction is made between names, addresses and routes. A *name* indicates an object to be sought. An *address* indicates the location of the object. A *route* indicates how to arrive at the object. The Internet protocol deals primarily with addresses. It is the task of higher level protocols (such as host-to-host or application) to make the mapping from names to addresses. The Internet module maps Internet addresses to local net addresses. It is the task of lower level procedures (such as local net or gateways) to make the mapping from local net addresses to routes.

Options
Options may or may not appear in datagrams. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation. In some environments, the security option may be

required in all datagrams.

The option field is variable in length. There may be zero or more options. There are two possible formats for an option:

- A single octet of option type.
- An option type octet, an option length octet and the actual option data octets.

The length octet includes the option type octet and the actual option data octets.

The option type octet has 3 fields:

1 bit: Copied flag. Indicates that this option is copied into all fragments during fragmentation:

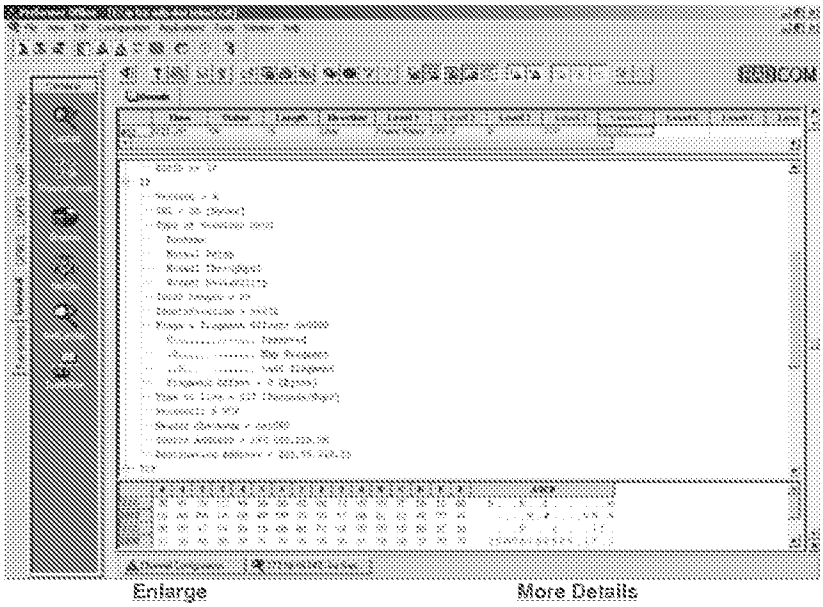
0 Copied.
1 Not copied.

2 bits: Option class

0 Control.
1 Reserved for future use.
2 Debugging and measurement.
3 Reserved for future use.

5 bits: Option number.

Data
IP data or higher layer protocol header.



Enlarge More Details

Interested in more details about testing this protocol? Click here

IPv6

RFC1883 http://www.cis.ohio-state.edu/htbin/rfc/rfc1883.html
RFC1827 http://www.cis.ohio-state.edu/htbin/rfc/rfc1827.html

IP version 6 (IPv6) is a new version of the Internet Protocol based on IPv4. IPv4 and IPv6 are demultiplexed at the media layer. For example, IPv6 packets are carried over Ethernet with the content type 86DD (hexadecimal) instead of IPv4's 0800.

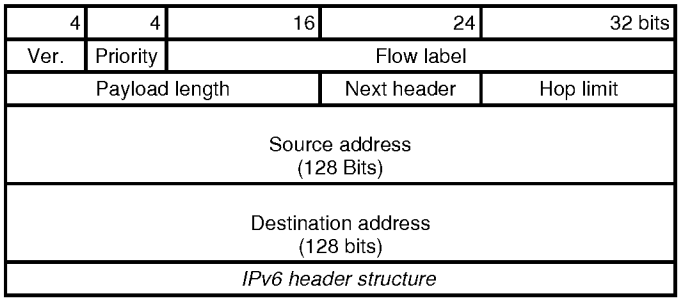
IPv6 increases the IP address size from 32 bits to 128 bits, to support more levels of

addressing hierarchy, a much greater number of addressable nodes and simpler auto-configuration of addresses. Scalability of multicast addresses is introduced. A new type of address called an *anycast address* is also defined, to send a packet to any one of a group of nodes.

Improved support for extensions and options - IPv6 options are placed in separate headers that are located between the IPv6 header and the transport layer header. Changes in the way IP header options are encoded allow more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future. The extension headers are: Hop-by-Hop Option, Routing (Type 0), Fragment, Destination Option, Authentication, Encapsulation Payload.

Flow labeling capability - A new capability has been added to enable the labeling of packets belonging to particular traffic flows for which the sender requests special handling, such as non-default Quality of Service or real-time service.

The IPv6 header structure is as follows:



IPv6 header structure

Version
Internet Protocol Version number (IPv6 is 6).

Priority
Enables a source to identify the desired delivery priority of the packets. Priority values are divided into ranges: traffic where the source provides congestion control and non-congestion control traffic.

Flow label
Used by a source to label those products for which it requests special handling by the IPv6 router. The flow is uniquely identified by the combination of a source address and a non-zero flow label.

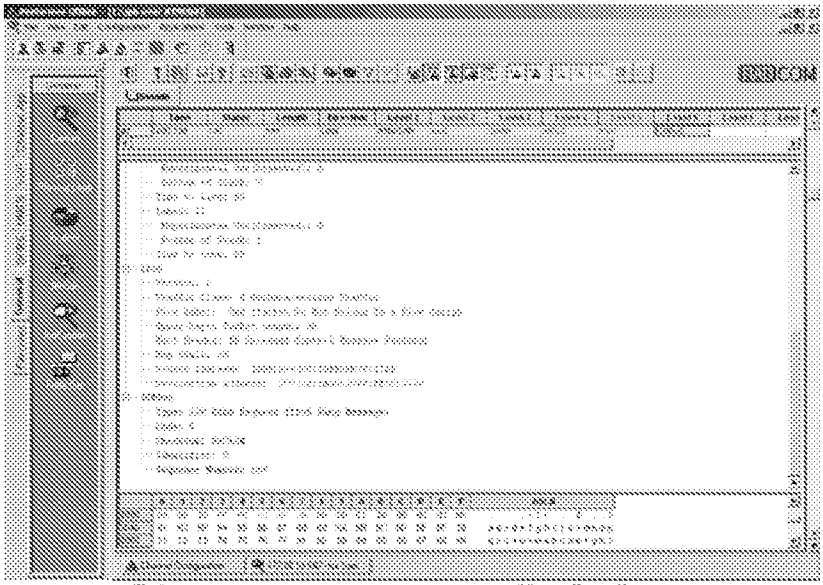
Payload length
Length of payload (in octets).

Next header
Identifies the type of header immediately following the IPv6 header.

Hop limit
8-bit integer that is decremented by one by each node that forwards the packet. The packet is discarded if the Hop Limit is decremented to zero.

Source address
128-bit address of the originator of the packet.

Destination address
128-bit address of the intended recipient of the packet.



Enlarge More Details

Interested in more details about testing this protocol? Click here

TCP

RFC793 http://www.cis.ohio-state.edu/htbin/rfc/rfc793.html
RFC1146 http://www.cis.ohio-state.edu/htbin/rfc/rfc1146.html
RFC1072 http://www.cis.ohio-state.edu/htbin/rfc/rfc1072.html
This RFC has been replaced by RFC 1323.
The information on this page will be updated to suit the new RFC in the near future.
RFC1693 http://www.cis.ohio-state.edu/htbin/rfc/rfc1693.html

IETF RFC793 defines the Transmission Control Protocol (TCP). TCP provides a reliable stream delivery and virtual connection service to applications through the use of sequenced acknowledgment with retransmission of packets when necessary.

The TCP header structure is as follows:

<table>
<tr><td colspan="8">16</td><td>32 bits</td></tr>
<tr><td colspan="8">Source port</td><td>Destination port</td></tr>
<tr><td colspan="9">Sequence number</td></tr>
<tr><td colspan="9">Acknowledgement number</td></tr>
<tr><td>Offset</td><td>Resrvd</td><td>U</td><td>A</td><td>P</td><td>R</td><td>S</td><td>F</td><td>Window</td></tr>
<tr><td colspan="8">Checksum</td><td>Urgent pointer</td></tr>
<tr><td colspan="9">Option + Padding</td></tr>
<tr><td colspan="9">Data</td></tr>
<tr><td colspan="9">TCP header structure</td></tr>
</table>

Source port
Source port number.

Destination port
Destination port number.

Sequence number
The sequence number of the first data octet in this segment (except when SYN is present). If SYN is present, the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1.

Acknowledgment number

If the ACK control bit is set, this field contains the value of the next sequence number which the sender of the segment is expecting to receive. Once a connection is established, this value is always sent.

Data offset
4 bits. The number of 32-bit words in the TCP header, which indicates where the data begins. The TCP header (even one including options) has a length which is an integral number of 32 bits.

Reserved
6 bits. Reserved for future use. Must be zero.

Control bits
6 bits. The control bits may be (from right to left):

U (URG)	Urgent pointer field significant.
A (ACK)	Acknowledgment field significant.
P (PSH)	Push function.
R (RST)	Reset the connection.
S (SYN)	Synchronize sequence numbers.
F (FIN)	No more data from sender.

Window
16 bits. The number of data octets which the sender of this segment is willing to accept, beginning with the octet indicated in the acknowledgment field.

Checksum
16 bits. The checksum field is the 16 bit one's complement of the one's complement sum of all 16-bit words in the header and text. If a segment contains an odd number of header and text octets to be checksummed, the last octet is padded on the right with zeros to form a 16-bit word for checksum purposes. The pad is not transmitted as part of the segment. While computing the checksum, the checksum field itself is replaced with zeros.

Urgent Pointer
16 bits. This field communicates the current value of the urgent pointer as a positive offset from the sequence number in this segment. The urgent pointer points to the sequence number of the octet following the urgent data. This field can only be interpreted in segments for which the URG control bit has been set.

Options
Options may be transmitted at the end of the TCP header and always have a length which is a multiple of 8 bits. All options are included in the checksum. An option may begin on any octet boundary.

There are two possible formats for an option:

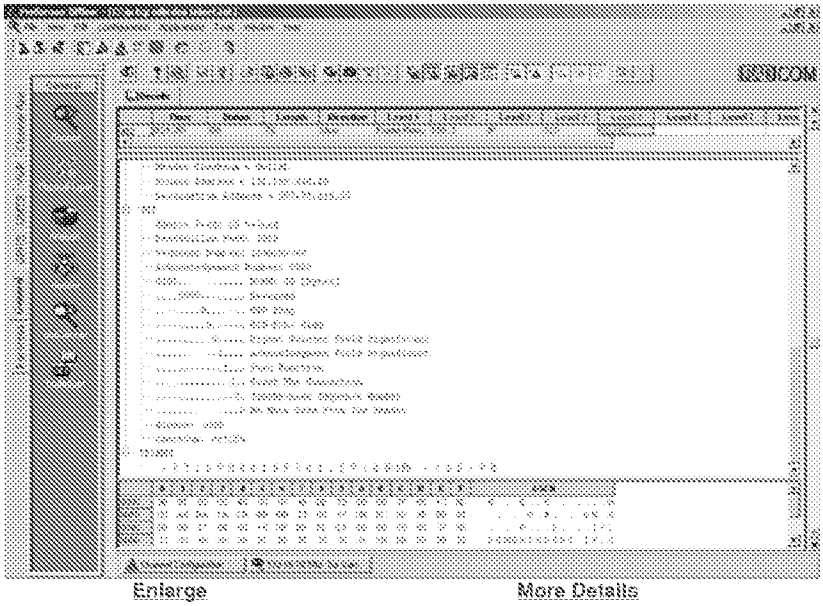
- A single octet of option type.
- An octet of option type, an octet of option length, and the actual option data octets.

The option length includes the option type and option length, as well as the option data octets.

The list of options may be shorter than that designated by the data offset field because the contents of the header beyond the End-of-Option option must be header padding i.e., zero.

A TCP must implement all options.

Data
TCP data or higher layer protocol.



Enlarge More Details

Interested in more details about testing this protocol? Click here

UDP

RFC768 http://www.cis.ohio-state.edu/htbin/rfc/rfc768.html

The User Datagram Protocol (UDP), defined by IETF RFC768, provides a simple, but unreliable message service for transaction-oriented services. Each UDP header carries both a source port identifier and destination port identifier, allowing high-level protocols to target specific applications and services among hosts.

The UDP header structure is shown as follows:

16	32 bits
Source port	Destination port
Length	Checksum
Data	

UDP header structure

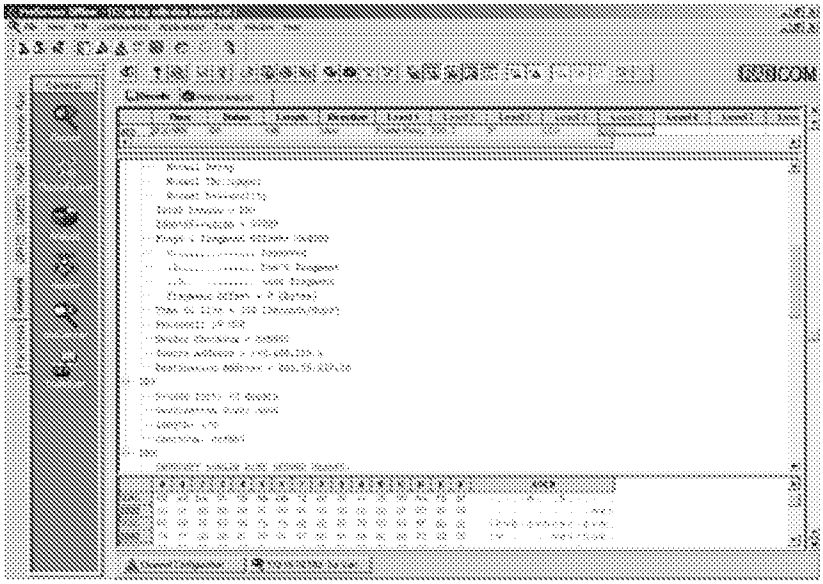
Source port
Source port is an optional field. When used, it indicates the port of the sending process and may be assumed to be the port to which a reply should be addressed in the absence of any other information. If not used, a value of zero is inserted.

Destination port
Destination port has a meaning within the context of a particular Internet destination address.

Length
The length in octets of this user datagram, including this header and the data. The minimum value of the length is eight.

Checksum
The 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header and the data, padded with zero octets at the end (if necessary) to make a multiple of two octets.

Data
UDP data field.



Enlarge More Details

Interested in more details about testing this protocol? Click here

1 2 3 4 5 6 7 8 9

TCP/IP Family Protocol Information

AH | ATMP | ARP/RARP | BGMP | BGP-4 | COPS | DCAP | DHCP | Diameter | DIS | DNS | DVMRP | EGP | EIGRP | ESP | FANP | Finger | FTP | HSRP | HTTP | ICMP/ICMPv6 | IGMP | IGRP | IMAP4 | IMPPpre/IMPPmes | IPDC | IP | IPv6 | IRC | ISAKMP | ISAKMP/IKE | iSCSI | ISTP | ISP | LDAP | L2F | L2TP | MARS | Mobile IP | MZAP | NARP | NetBIOS/IP | NHRP | NTP | OSPF | PIM | POP3 | PPTP | Radius | RLOGIN | RIP2 | RIPng for IPv6 | RSVP | RTSP | RUDP | SCTP | S-HTTP | SLP | SMTP | SNMP | SOCKS | TACACS+ | TALI | TCP | TELNET | TFTP | TLS | TRIP | UDP | Van Jacobson | VRRP | WCCP | X-Window | XOT

Additional Information

- Protocol Testing - Products
- Protocol Testing - Solutions
- Protocol Testing - Technologies